UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/697,890	10/30/2003	Gail A. Alverson	32475-8003US6	9191

25096   7590   01/23/2009
PERKINS COIE LLP
PATENT-SEA
P.O. BOX 1247
SEATTLE, WA 98111-1247

EXAMINER
ZHEN, LI B

ART UNIT	PAPER NUMBER
2194	

MAIL DATE	DELIVERY MODE
01/23/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

<table>
<tr><td rowspan="2">Office Action Summary</td><td>Application No.
10/697,890</td><td>Applicant(s)
ALVERSON ET AL.</td></tr>
<tr><td>Examiner
LI B. ZHEN</td><td>Art Unit
2194</td></tr>
</table>

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 October 2008*.
2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13,16,18-28,31,33,36,38-47,49 and 51-54* is/are pending in the application.
  4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5)☒ Claim(s) *41-44 and 54* is/are allowed.
6)☒ Claim(s) *1-13,16,18-28,31,33,36,38-40,45-47,49 and 51-53* is/are rejected.
7)☐ Claim(s) _____ is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.
10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.
  Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
  Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
  a)☐ All   b)☐ Some * c)☐ None of:
    1.☐ Certified copies of the priority documents have been received.
    2.☐ Certified copies of the priority documents have been received in Application No. _____.
    3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
  * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)
1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
   Paper No(s)/Mail Date *7/9/08;8/8/08;1/9/09*.
4) ☐ Interview Summary (PTO-413)
   Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1.   Claims 1-13, 16, 18-28, 31, 33, 36, 38-47, 49 and 51-54 are pending in the

application.

Response to Arguments

2.   Applicant's arguments with respect to the claims have been considered but are

moot in view of the new ground(s) of rejection.

Allowable Subject Matter

3.   Claims 41 – 44 and 54 are allowed.

Claim Rejections - 35 USC § 103

4.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5.   **Claims 1-13, 16, 18-28, 31, 33, 36, 38-40, 45-47, 49 and 51-53 are rejected**

under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,341,301 to

Hagan [previously cited] in view of U.S. Patent No. 6,209,003 to Mattis et al.

[hereinafter Mattis, previously cited] and further in view of U.S. Patent No.

6,115,802 to Tock et al. [hereinafter Tock].

6.   As to claim 1, Hagan teaches a method in a computer system for accessing a

collection of data items [queue 200; col. 4, lines 10 – 25], the method comprising:

when a thread adds data item to the collection, fetching and adding to a write

counter [After a second item is posted to queue 200…write pointer 208 now points to

element 3; col. 4, lines 25 – 40 and col. 4, lines 40 – 56], the fetched write counter [write

pointer 208 points to the element (in the "last" position) in which data is being written

into; col. 4, lines 18 – 25] pointing to a bucket within a bucket array [col. 3, line 62 – col.

4, line 10], each bucket in the bucket array corresponding to a portion of the collection

of data items [col. 3, line 62 – col. 4, line 10];

reading from the bucket pointed to by the fetched write counter using a

synchronization access mode of sync [only one queue is being processed while others

remained locked; col. 8, line 57 – col. 9, line 14];

after reading the bucket [queue entry that has been read is marked as empty;

col. 7, lines 25 – 50], storing the data item in association with the bucket pointed to by

the fetched write counter [queue handling routine 308 reads the first queue entry in

queue A, decodes the proper processing context for queue A; col. 5, lines 19 – 33];

writing to the bucket pointed to by the fetched write counter using a

synchronization access mode of sync [col. 8, line 57 – col. 9, line 20]. Hagan does not

specifically disclose if the bucket is currently marked empty, the reading is delayed until

the bucket is marked full and so that the bucket is marked empty upon reading to

prevent subsequent reading from the bucket until the bucket is written to, writing to the

bucket pointed to by the fetched write counter using a synchronization access mode of

sync so that the bucket is marked full to allow reading from the bucket, fetching and adding to a lower bound to indicate the number of data items added to the collection.

However, Mattis teaches accessing a collection of data items [col. 7, lines 31 – 54] by a plurality of threads [col. 36, lines 13 – 23], if the bucket is currently marked empty [two one-bit values that indicate whether the corresponding arena is empty; col. 17, lines 48 – 55], the reading is delayed until the bucket is marked full and so that the bucket is marked empty upon reading to prevent subsequent reading from the bucket until the bucket is written to [If the matching block is currently in the process of being created or destroyed, then in step 840 an error message is returned to the protocol engine 70 indicating that the current block is not available; col. 33, lines 13 – 20], writing to the bucket pointed to by the fetched write counter using a synchronization access mode of sync [a lock is obtained; col. 34, lines 1 – 26] so that the bucket is marked full allowing reading from the bucket [write count is incremented; col. 34, lines 36 – 46], fetching and adding to a lower bound to indicate the number of data items added to the collection [sum of all the counter values; col. 27, lines 20 – 55].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the invention of Hagan to include the features of Mattis. One would have been motivated to make the combination because this provides high performance, as measured by low latency and high throughput for object store operations, and large numbers of concurrent operations [col. 38, lines 26 – 35 of Mattis]. Hagan and Mattis do not teach wherein multiple threads can be simultaneously accessing different buckets.

However, Tock teaches multiple threads can be simultaneously accessing different buckets [a hash table is available for concurrent access by several independently operating processes or threads; col. 8, lines 30 – 36, col. 1, lines 56 – 65 and col. 2, lines 36 – 40].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to further modify the invention of Hagan and Mattis to incorporate the features of Tock. One of ordinary skill in the art would have been motivated to make the combination because provides lockless lookup operations to occur concurrently with either one insert or one delete operation [col. 8, line 63 – col. 9, line 19 of Tock].

7.   As to claim 18, Hagan as modified teaches a method in a computer system for accessing a collection of data items [col. 4, lines 10 – 25 of Hagan], the method comprising:

when removing a data item from the collection [read from queue 200; col. 4, lines 17 – 25 of Hagan], fetching and adding to a read counter [col. 4, lines 25 – 40 of Hagan], the fetched read counter pointing to a bucket within a bucket array [read pointer 206 is employed to point to the element (in the "first" position) currently being read from queue 200; col. 4, lines 16 – 25 of Hagan], each bucket in the bucket array corresponding to a portion of the collection of data items [col. 3, line 62 – col. 4, line 10 of Hagan], wherein data items associated with different buckets can be accessed simultaneously [col. 8, lines 30 – 36, col. 1, lines 56 – 65 and col. 2, lines 36 – 40 of Tock];

reading from the bucket pointed to by the fetched read counter using a

synchronization access mode of sync [col. 8, line 57 – col. 9, line 14 of Hagan] so that if

the bucket is currently marked empty [two one-bit values that indicate whether the

corresponding arena is empty; col. 17, lines 48 – 55 of Mattis], the reading is delayed

until the bucket is marked full and so that the bucket is marked empty upon reading to

prevent subsequent reading from the bucket until the bucket is written to [If the

matching block is currently in the process of being created or destroyed, then in step

840 an error message is returned to the protocol engine 70 indicating that the current

block is not available; col. 33, lines 13 – 20 of Mattis];

removing the data item from association with the bucket pointed to by the fetched

read counter [queue entry that has been read is marked as empty (step 506); col. 7,

lines 24 – 50 of Hagan]; and

writing to the bucket pointed to by the fetched write counter using a

synchronization access mode of sync [col. 8, line 57 – col. 9, line 20 of Hagan] so that

the bucket is marked full to allow reading from the bucket [col. 34, lines 36 – 46 of

Mattis].

8.   As to claim 33, Hagan as modified teaches a method in a computer system for

accessing a collection of data items [queue 200; col. 4, lines 10 – 25 of Hagan], the

method comprising:

defining a pointer to indicate a location for a bucket within a bucket array [read

pointer 206; col. 4, lines 16 – 25 of Hagan], each bucket in the bucket array

corresponding to a portion of the collection of data items [col. 3, line 62 – col. 4, line 10

of Hagan], wherein multiple readers and writers can be accessing data items of different

buckets simultaneously [col. 8, lines 30 – 36, col. 1, lines 56 – 65 and col. 2, lines 36 –

40 of Tock];

defining a lower bound to indicate a number of items in the collection [sum of all

the counter values; col. 27, lines 20 – 55 of Mattis]; and

determining based on the lower bound whether the collection has a data item

[col. 7, lines 24 – 50 of Hagan]; and

when it is determined that the collection has a data item, adjusting the lower

bound to indicate that the collection has one less data item [col. 27, lines 31 – 55 of

Mattis];

reading from the bucket at the location indicated by the pointer [col. 7, lines 24 –

50 of Hagan] using a synchronization access mode of sync [col. 8, line 57 – col. 9, line

14 of Hagan] so that if the bucket is currently marked empty [two one-bit values that

indicate whether the corresponding arena is empty; col. 17, lines 48 – 55 of Mattis], the

reading is delayed until the bucket is marked full and so that the bucket is marked

empty upon reading to prevent subsequent reading from the bucket until the bucket is

written to [If the matching block is currently in the process of being created or destroyed,

then in step 840 an error message is returned to the protocol engine 70 indicating that

the current block is not available; col. 33, lines 13 – 20 of Mattis];

removing the data item from association with the bucket pointed to by the pointer [queue entry that has been read is marked as empty (step 506); col. 7, lines 24 – 50 of Hagan]; and

writing to the bucket at the location defined by the pointer using a synchronization access mode of sync [col. 8, line 57 – col. 9, line 20 of Hagan] so that the bucket is marked full to allow reading from the bucket [col. 34, lines 36 – 46 of Mattis].

9.   As to claim 49, Hagan as modified teaches a computer system for accessing data [queue 200; col. 4, lines 10 – 25 of Hagan], comprising:

a collection of buckets [col. 4, lines 10 – 25 of Hagan and col. 7, lines 31 – 54 of Mattis], each bucket corresponding to a portion of a collection of data items [col. 3, line 62 – col. 4, line 10 of Hagan], wherein multiple readers and writers can access the data items of different buckets in parallel [col. 8, lines 30 – 36, col. 1, lines 56 – 65 and col. 2, lines 36 – 40 of Tock];

a read counter [read pointer; col. 4, lines 25 – 40 of Hagan] and a write counter [write pointer; col. 4, lines 18 – 25 of Hagan] that point to buckets within the collection [col. 4, lines 18 – 25 of Hagan];

a data structure for each bucket with locations for holding data at each of the buckets [col. 4, lines 10 – 25 of Hagan and col. 7, lines 31 – 54 of Mattis]; and

a lower bound indicating whether a data item is currently stored in the data structure [col. 27, lines 20 – 55 of Mattis];

a component that adjusts the lower bound to indicate a change in the number data items [col. 27, lines 31 – 55 of Mattis];

a component that reads from the bucket at the location pointer to by the counter [col. 7, lines 24 – 50 of Hagan] using a synchronization access mode of sync [col. 8, line 57 – col. 9, line 14 of Hagan] so that if the bucket currently marked empty [two one-bit values that indicate whether the corresponding arena is empty; col. 17, lines 48 – 55 of Mattis], the reading is delayed until the bucket is marked full and so that the bucket is marked empty upon reading to prevent subsequent reading from the bucket until the bucket is written to [col. 33, lines 13 – 20 of Mattis];

a component that modifies data items associated with the bucket at the location pointer to by the pointer [col. 7, lines 24 – 50 of Hagan]; and

a component that writes to the bucket at the location pointer to by the pointer using a synchronization access mode of sync [col. 8, line 57 – col. 9, line 20 of Hagan] so that the bucket is marked full to allow reading from the bucket [col. 34, lines 36 – 46 of Mattis].

10.   As to claim 2, Hagan as modified teaches the bucket pointed to by the fetched write counter contains a pointer to a linked list of data items [col. 16, lines 35 – 45 of Mattis].

11.   As to claim 3, Hagan as modified teaches the fetched write counter modulo a number of buckets in the bucket array points to a bucket within the bucket array [col. 34, lines 10 – 26 of Mattis].

12.   As to claim 4, Hagan teaches adding adds one to the write counter [col. 4, lines 25 – 40].

13.   As to claim 5, Hagan as modified teaches the adding adds a size of a bucket to the write counter [col. 34, lines 36 – 46 of Mattis].

14.   As to claim 6, Hagan teaches when a thread removes a data item from the collection [col. 4, lines 17 – 25], fetching and adding to a read counter [col. 4, lines 25 – 40], the fetched read counter pointing to a bucket within the bucket array [col. 4, lines 16 – 25]; reading from the bucket pointed to by the fetched read counter using a synchronization access mode of sync [col. 8, line 57 – col. 9, line 14]; removing the data item from association with the bucket pointed to by the fetched read counter [col. 7, lines 24 – 50]; and writing to the bucket pointed to by the fetched write counter using a synchronization access mode of sync [col. 8, line 57 – col. 9, line 20].

15.   As to claim 7, Hagan as modified teaches when a thread removes a data item from the collection, checking the lower bound to ensure that the collection contains a data item [col. 27, lines 20 – 55 of Mattis], when it cannot be ensured that the collection

contains a data item [col. 7, lines 24 – 50 of Hagan], indicating that the collection may

be empty [col. 7, lines 24 – 50 of Hagan]; and when it can be ensured that the collection

contains a data item [col. 34, lines 1 – 46 of Mattis], fetching and adding to a read

counter [col. 4, lines 25 – 40 of Hagan], the fetched read counter pointing to a bucket

within the bucket array [col. 4, lines 16 – 25 of Hagan]; reading from the bucket pointed

to by the fetched read counter using a synchronization access mode of sync [col. 8, line

57 – col. 9, line 14 of Hagan]; removing the data item from association with the bucket

pointed to by the fetched read counter [col. 7, lines 24 – 50 of Hagan]; and writing to the

bucket pointed to by the fetched write counter using a synchronization access mode of

sync [col. 8, line 57 – col. 9, line 20 of Hagan].

16.   As to claim 8, Hagan as modified teaches the checking includes fetching and

adding a negative number to the lower bound [col. 35, lines 52 – 61 of Mattis].

17.   As to claim 9, Hagan as modified teaches the checking includes fetching and

adding a positive number to the lower bound when it cannot be ensured that the

collection contains an item [col. 26, lines 7 – 23 of Mattis].

18.   As to claim 10, Hagan teaches the synchronization access mode of sync

prevents simultaneous access of a bucket by multiple threads [col. 8, line 57 – col. 9,

line 14].

19.   As to claim 11, Hagan as modified teaches the collection of data items is stored in an array data structure [col. 10, lines 56 – 65 of Mattis].

20.   As to claim 12, Hagan as modified teaches the collection of data items is stored in a linked list data structure [col. 4, lines 10 – 19 of Hagan and col. 16, lines 35 – 45 of Mattis].

21.   As to claim 13, Hagan as modified teaches the collection of data items is stored in a tree data structure [col. 16, lines 35 – 45 of Mattis].

22.   As to claim 16, Hagan teaches the writing is permitted only when the bucket is empty [col. 8, line 57 – col. 9, line 20].

23.   As to claim 19, Hagan as modified teaches before fetching and adding to the read counter checking a lower bound [col. 34, lines 36 – 46 of Mattis] to ensure that the collection contains a data item [col. 7, lines 24 – 50 of Hagan].

24.   As to claim 20, Hagan teaches wherein it cannot be ensured that the collection contains a data item, indicating that a data item cannot be removed [col. 7, lines 24 – 50].

25.   As to claim 21, Hagan teaches the bucket pointed to by the fetched read counter contains a pointer to a linked list of data items [col. 4, lines 10 – 19].

26.   As to claim 22, Hagan as modified teaches the fetched read counter modulo a number of buckets in the bucket array points to a bucket within the bucket array [col. 10, lines 56 – 65 of Mattis].

27.   As to claim 23, Hagan teaches the removing adds one to the read counter [col. 4, lines 25 – 40].

28.   As to claim 24, Hagan as modified teaches the adding adds a size of a bucket to the read counter [col. 34, lines 36 – 46 of Mattis].

29.   As to claim 25, Hagan teaches the synchronization access mode of sync prevents simultaneous access of a bucket by multiple threads [col. 8, line 57 – col. 9, line 14].

30.   As to claim 26, Hagan as modified teaches the collection of data items is stored in an array data structure [col. 10, lines 56 – 65 of Mattis].

31.   As to claim 27, Hagan as modified teaches the collection of data items is stored in a linked list data structure [col. 4, lines 10 – 19 of Hagan and col. 16, lines 35 – 45 of Mattis].

32.   As to claim 28, Hagan as modified teaches the collection of data items is stored in a tree data structure [col. 16, lines 35 – 45 of Mattis].

33.   As to claim 31, Hagan teaches the writing is permitted only when the bucket is empty [col. 8, line 57 – col. 9, line 20].

34.   As to claim 36, Hagan teaches the write access to the location is permitted only when the location is empty [col. 8, line 57 – col. 9, line 20].

35.   As to claim 38, Hagan teaches wherein the data items of the collection are accessed by multiple readers and writers [col. 8, line 57 – col. 9, line 14].

36.   As to claim 39, Hagan teaches the data items of the collection are accessed by multiple producers [col. 5, lines 5 – 20].

37.   As to claim 40, Hagan teaches wherein the data items of the collection are accessed by multiple consumers [col. 5, lines 5 – 20].

38.   As to claim 45, Hagan teaches wherein each bucket of the array of buckets

includes a pointer to data items [col. 8, line 57 – col. 9, line 14].

39.   As to claim 46, Hagan teaches wherein the collection includes a write counter

that indicates a bucket into which a next data item is to be stored [col. 4, lines 18 – 25]

and a read counter that indicates a bucket from which a next data item is to be read

[col. 4, lines 25 – 40].

40.   As to claim 47, Hagan as modified teaches wherein the pointed to data items are

stored in a linked list [col. 4, lines 10 – 19 of Hagan and col. 16, lines 35 – 45 of Mattis].

41.   As to claim 51, Hagan teaches wherein write access to a bucket is permitted only

when the bucket is empty [col. 7, lines 24 – 50].

42.   As to claim 52, Hagan teaches accessing programs that each operates in a

different thread [col. 5, lines 5 – 20].

43.   As to claim 53, Hagan teaches wherein the data is accessed by multiple reading

threads and writing threads [col. 5, lines 5 – 20].

Conclusion

44.   Applicant's amendment necessitated the new ground(s) of rejection presented in

this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37

CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the date of this final action.

CONTACT INFORMATION

45.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768.

The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Meng-Ai An can be reached on (571)272-3756. The fax phone number for

the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Li B. Zhen/   Li B. Zhen
Primary Examiner, Art Unit 2194   Primary Examiner
                                                            Art Unit 2194